C. Remarks

In the office action, Claim 1 was rejected under 35 U.S.C. 112 (first and second paragraphs). Claims 1, 12, 20, and 36 were also rejected under 35 U.S.C. 101, as being directed to a non-statutory subject matter. Claims 1-5, 7-16, 20-24, 27-40, and 44-48 were rejected under 35 U.S.C. 102 (e), as anticipated by the patent to Abrams et al. U.S patent application publication number: 2002/0166117 A1. Claims 1-5, 7-16, 20-24, 27-40, and 44-48 were also rejected under 35 U.S.C. 102 (e), as anticipated by the patent to Lanzillo, JR. et. al. U.S patent application publication number: 2002/0032602 A1.

The specification

The specification has been carefully reviewed and, contrary to the statements in the office action, no substantive errors were found, thus no corrections have been made at this time. However if the examiner notes any specific errors that require correction they will be made in a further amendment.

Rejections under 35 U.S.C. 101, and 35 U.S.C. 112:

With respect to the rejections of claims 1, 12, 20, and 30 under 35 U.S.C. 101, the Office Action states that the claims do not recite the use of a computer to execute the claimed method. This rejection is not understood. For example, Claim 1 (as well as claims 12, 20, and 30) clearly state that the applications are *running on a computer*, and the computer is connected in a *network of computers*, which include a host program including one or more symbionts. The host program and the one or more symbionts then carry out the steps of the method. As the host program and the symbionts are *running on a*

computer, the claims are clearly directed to carrying out the method by *using the computers*. Thus, the application and the claims relate to a technological art and the rejection under 35 U.S.C. 101 is not well founded.

Regarding the rejection under 35 U.S.C. 112 (first paragraph) of claim 1 that the recitation: "*said host has not been redirected more than a predetermined number of times*" was not enabled, this language has been removed in order to overcome the rejection. Further, the insufficient antecedent basis in Claim 1 has been rectified in order to remove the rejection under 35 U.S.C.112 (second paragraph).

Prior Art Rejections

The present invention provides a system and a method for handling a request for a resource in an application running on a computer. The computer includes a plurality of resources. The resources are encapsulated in symbionts. A symbiont is a self-replicating program that is run on host programs. These host programs are run on computers that are connected to a network. Applications running on a computer send a request (hereinafter referred to as a requesting host) for a resource to the host program. The host program running on the computer receives the request (herein after referred to as a receiving host) and contacts a symbiont encapsulating the resource. The symbiont either serves the request, replicates the resource on the requesting host, or redirects the request to a connected replicate, based on a predetermined criterion. The replicated resources, encapsulated in symbionts, are connected together to form a multiply connected ring. A newly replicated symbiont, encapsulating the resource, connects to the multiply connected ring. Each symbiont checks its own load at regular time intervals. If the load of the

symbiont is less than a predetermined threshold, it is eliminated from the network. At least one of the symbionts in the multiply connected ring is marked as immortal.

In contrast, Abrams teaches a system and a method for processing a computer application with dynamic capacity control. Abrams teaches a system and a method for pay-per-use charging on-demand basis. Further, Abrams teaches a system that provides on-demand, scalable computational resources to application providers in a distributed network. The application providers are capable of monitoring, updating and replacing the distributed network. The on-demand system includes computers and/or servers that are installed globally.

The system also includes an application-switching architecture, an edgepoint, real time processing engines, and a plurality of server resources with target multimedia content. The application-switching architecture includes an application snapshot or appshot. The appshot is a set of all the data and/or state necessary to halt (and then restore and restart) at least one application at a given point of time, such that the application may be restored at a later time on substantially any machine. An edgepoint is capable of operating as a server. Further, the edgepoint comprises one or more servers such as the Sun server from Sun Microsystems.

The on-demand system is capable of relocating or replicating a resource (an appshot) to other or alternative sets of computational systems, such as other computational modules and/or other edgepoints. Once an appshot is replicated in the distributed network, a user can be routed to the most optimal (nearest) edgepoint with a specified application, instead of being routed to the central site, with the help of a Global Dispatcher (GD). The resources needed are dynamically available on demand, and can

be freed on/from application servers when they are not required.

Lanzillo teaches a system and a method for the distribution of target advertisements over a communication network, such as by an email to a set of users or subscribers. Further, Lanzillo teaches a system for recipient selection and message delivery, whereby demographic information is updated in time, new database lists are quickly generated, and email advertisements targeted, sent and tracked. The system also includes features including scalability and fault tolerance.

The entire system can run on a single physical machine, or, with virtually unlimited scalability, spread across multiple machines in a Local Area Network (LAN), for better performance and availability.

A Master Controller (MC) agent runs as an application on windows 2000. The MC process runs on each server participating in the system, but only one MC is dynamically elected as the Primary Master Controller (PMC) of the entire system. Each of the other MCs in the system is subordinate to the PMC. The MC/ PMC is responsible for starting other agent processes when it detects an increase in the workload, and terminating agents when they are no longer needed, in an effort to balance the load of system resources. When a designated PMC process is stopped for any reason, one of the other MCs automatically takes over the PMC role. The system can be configured to automatically compensate for increased workloads by starting and stopping additional agents. The function of the PMC is to monitor systems operations and coordinate queue negotiation between all the messaging agents. This is carried out through a listener queue, where all the messaging agents make requests for their queue designations. When the PMC detects an increased workload state, indicated by a work queue

exceeding a specified queue length, it automatically redirects the process of sending agent start requests to other MCs in the system. Each machine participating in the system maintains a profile. The profile stores information pertaining to the number of instances that need to be maintained according to the size and speed of the machine. The PMC uses this information, along with current machine state, to determine which MC, running on a particular machine, is to be asked to start additional agents. The system capacity can be dynamically increased by adding a new machine to the LAN and starting an MC on it. The new MC registers its presence with the system PMC and is immediately available for new agent start requests.

Independent claim 1

In order to more clearly define, and distinctly claim the present invention from that of Abrams's and Lanzillo's patent, independent claim 1 has been amended to recite that the method of the present invention relates to handling a request for resources in a network of computers. In view of the Examiner's rejection of Claim 1 under 35 U.S.C. 112, the statement "*said host has not been redirected more than a predetermined number of times*" has been removed in order to overcome the rejection. Claim 1 has been further amended by incorporating the statement "*the one or more symbionts being connected to each other for communication"* and thus Claim 4 has been cancelled. Further, the insufficient antecedent basis in Claim 1 has been rectified by this amendment in order to remove the rejection under 35 U.S.C.112.

According to the method of the present invention, as set forth in amended claim 1, the present invention may be practiced on any type of network architecture between

computers. However, Abrams's invention is practiced primarily on a client-server architecture. In the present invention, each computer of a network includes a host program, and the host program may include a self- replicating program known as a symbiont. Hence, the present invention eliminates the requirement of any dedicated hardware to replicate a resource. However, Abrams's invention teaches the need for dedicated hardware such as an edge point. The edge point includes a plurality of servers. Further, Lanzillo teaches the need for an MC, which runs only in windows 2000 and can be practiced in a Local Area Network (LAN).

The symbiont of the present invention is a self-replicating program that encapsulates a resource such as a computational resource, data or application. The symbiont either serves the request, replicates the resource on the requesting host, or redirects the request to a connected replicate, based on a predetermined criterion. The replication takes place if the load on the symbiont exceeds a threshold l_{max}. Alternatively, the symbiont replicates itself on the requesting host if the load on the symbiont exceeds l_{max} and the load on the neighboring symbionts that encapsulate the resource is greater than a threshold t. Hence, the symbiont of the present invention is an intelligent decision-making component that is responsible for resource distribution. The support for these recitations is found on page 12, lines 9-25 and page 13, lines 1-9. It may be noted that as per Abrams's invention, the application-switching architecture is the decision-making component that manages the distributed resources (end and system resources). Further, it may be noted that as per Lanzillo's invention, the PMC is the decision- making component that manages the distributed resources.

According to the present invention, the replication of a symbiont can occur on a

requesting host. Abrams teaches a method of replication of an object, the replication of an object occurs only on interconnected edgepoints and/or on interconnected servers. The edgepoint includes dedicated hardware such as servers. Moreover, Abrams and Lanzillo do not teach that a resource can be replicated on a requesting client. Further, Lanzillo teaches that the method of replication of the resource occurs only on a system/ server that are connected to the LAN, thereby necessitating the requirement of dedicated hardware. Therefore, it is apparent that the present invention offers more flexibility and robustness for handling a request. Further, the present invention was implemented without any dedicated hardware, since both the symbiont and the host are functions in the form of software components. The support for this recitation is found on page 10, lines 13-15.

According to the present invention, all the replicates of a particular resource are connected together. The connection between the symbionts encapsulating the resource in the network provides load information about neighboring symbionts (hereinafter referred to as connected replicates). This reduces the time taken by a symbiont to make a decision about handling a request, for example, if the symbiont retains the information about the connected replicates, and redirects the request to a connected replicate with the least load. The support for these recitations is found on page 11, lines 9-24. However, Abrams does not teach any connection between the object replicas, it teaches that thousands of servers in hundreds of edgepoints are globally deployed and linked to create a virtual single server view throughout the world. Further, Abrams teaches about the intelligence of the GD, to route the requests to an optimal edgepoint. Lanzillo teaches that a plurality of MCs are connected to create a LAN, wherein only one PMC is the decision-

making component routing user requests to other MCs that are interconnected in the LAN. Whereas, in the present invention, each symbiont is an independent decision-making component that redirects, replicates and exits from the multiply connected ring and can run in any type of network.

According to the present invention, the request is redirected to a connected replicate of the symbiont encapsulating the resource. The symbiont may be located on a computer anywhere in the network. The support for these recitations is found on page 12, lines 11-25 and page 13, lines 1-9. In the present invention, all the symbionts are intelligent units that handle requests for a resource. Lanzillo teaches the method for redirecting the request for the resource requested by the client to another MC. It may be noted that as per Abrams's invention, GD is the decision-making component for routing the user's requests to the optimal (nearest) edgepoint. Further, it may be noted that as per Lanzillo's invention, PMC is the decision-making component for routing the user's requests to other MC.

The host is a program that provides a suitable living environment for the symbiont to run, i.e. it provides memory, storage, script interpretation and other services necessary for the symbiont to function, whereby the symbiont runs within the host. According to Abrams, there is no such program that provides a living environment to an appshot. According to Lanzillo, there is no such program that provides a living environment to the MC/ PMC.

<u>Dependent claim 2, 13, 21, and 37</u>

As to claims 2, 13, 21, and 37 in the present invention, the computers in the network include host programs that expose the symbionts in the network to the applications running on the computer. The applications on the computer, therefore, have information pertaining to the resources present in the network. This enables the host to manage the process of communication with the symbionts in a more effective manner, since the host would know *a priori* about the symbionts that have the requisite resource. The support for this recitation is found on page 11, lines 2-3. Abrams teaches that application providers distribute applications in a network, to utilize the distributed computer resources for processing the applications.

Further, the computer in the network that includes the host program exposes the symbionts stored within the host on the network. This enables each host program in the network to obtain information regarding the symbionts present in the network. Therefore, this feature optimizes the time taken by the host program to find an appropriate symbiont that can serve its request. According to Abrams's invention, the application provider enables the distribution of applications in a network, to utilize the distributed computer resources for processing the applications. The application provider has the information required to enable a request from a client to get mapped on a server.

According to Lanzillo, an ad agency sends ads to individual subscribers through email in different formats, where the user needs to login PMC with the user-id and password, to send a request for the resource.

<u>Dependent claims 3, 14, 22, and 38</u> have been canceled without prejudice.

Dependent claims 4, 15, 23, and 39 have been canceled without prejudice.

Dependent claims 5, 16, 24, and 40

As to claims 5, 16, 24 and 40, in the present invention, all replicates of a particular resource are connected together in a multiply connected ring. The symbiont is an independent decision-making component that redirects, replicates and exits from the multiply connected ring. In the multiply connected ring, there is a tradeoff between the networking load and the resources load. This trade off depends on the number of neighbors to which each symbiont is connected. The support for these recitations is found on page 11, lines 9-24. However, Abrams uses a pre-existing Internet infrastructure to connect the edgepoints. Moreover, Abrams does not teach the concept of a multiply connected ring. Further, Abrams teaches that an application-switching software and a GD are needed to replicate a resource on another server, and redirect the request to an optimal edgepoint. Hence, the present invention provides the flexibility to alter the tradeoff between the networking and resources loads.

Dependent claims 7-9, 28-30

As to claims 7-9 and 28-30, by reducing the threshold l_{max} in the present invention, the number of symbionts in a network can be increased dynamically (according to the load on the symbiont in the network). For instance, if the requirement for a particular resource is high, the value of l_{max} may be decreased, so that a larger number of replicates of a symbiont may be created. Alternatively, the value of l_{max} may be kept higher, to minimize the number of possible replicates of the symbiont. Further, the value of the

threshold load, t, of the connected symbionts is always lower than that of l_{max}. The support for these recitations is found on page 12, lines 11-23 and page 13, lines 1-3. Such an alteration of the value of l_{max} occurs in real time and not on the basis of time multiplexing of the resource (target content of Abrams).

Further, the present invention teaches the use of two different thresholds. The value of t is deliberately kept lower than $l_{max,}$ since the load of the neighboring symbionts is acquired by the symbiont only after predefined time intervals. However, Abrams's and Lanzillo's inventions do not teach the use of two thresholds, l_{max} and t, or any relation between them. Moreover, the thresholds of l_{max} and t do not depend on the time multiplexing of the resource, as mentioned by Abrams.

Dependent claim 10

As to claim 10, the present invention recites that a symbiont will redirect a request to a connected replicate with the least load. The support for this recitation is found on page 12, lines 18-24. Therefore, according to the present invention, the symbiont has the intelligence to choose the connected replicate that is best suited to handle the request. This step decreases the time taken for the request to be handled, thereby increasing the speed with which it is handled. Hence, the present invention increases the efficiency of the request being handled.

Abrams teaches GD that handles the requests, based on factors such as the demand for target content (resource), the willingness of a server, the number of replicas available, and the resources available with each edgepoint, which makes the entire process dependent on the GD. Lanzillo teaches the PMC that handles the request, based

on factors such as the demand for the resource, the willingness of the server/ system, the number of replicas available, and the resources available in each system/ server, which makes the entire process dependent on the PMC. However, in the present invention, there are multiple copies of a component with intelligence (symbiont), which makes the system robust. Hence, the symbiont in the present invention is superior to the PMC/ MC and the GD.

Dependent claim 11 has been canceled without prejudice.

Independent claim 12

In order to more clearly define and distinctly claim the present invention from that of Abrams, independent claim 12 has been amended to recite that the system of the present invention relates to handling a request for a resource on a network of computers. According to the system of the present invention, as set forth in amended claim 12, the present invention may be practiced on various types of network architecture between computers. However, Abrams's invention is practiced primarily on a client-server architecture. In the present invention, each computer in a network comprises a host program. Further, the host program comprises a self-replicating program known as a symbiont. Therefore, the present invention does not use any dedicated hardware for replicating the resource. However, Abrams teaches the need for a dedicated hardware such as an edge point, the edge point comprising a plurality of servers. Lanzillo teaches the need for a dedicated hardware such as a server/ system that needs to be connected in a LAN.

The symbiont of the present invention is a self-replicating program that encapsulates the resource, such as a computational resource or data or an application. Abrams's object replica is a replica of the target content requested by a client. However, in the present invention, the symbiont handles the request for the resource by replicating itself on the requesting host. The replication takes place if the load of the symbiont exceeds a threshold l_{max}. Alternatively, the symbiont replicates on the requesting host if the load on the symbiont exceeds the threshold l_{max} and the load on the neighboring symbionts that encapsulate the resource is greater than that of the threshold t. Hence, the symbiont in the present invention is an intelligent decision-making component that is responsible for resource distribution. The support for these recitations is found on page 12, lines 9-25 and page 13, lines 1-8. Abrams teaches an application-switching architecture, which is a decision-making component that manages the distribution of the resources (end resources and system resources). Lanzillo teaches MC/ PMC, which is the decision-making component that manages the distribution of the resources (end resources and system resources).

According to the present invention, the replication of a resource can occur on the requesting host. However, Abrams teaches a method for the replication of an object. The replication of an object occurs on interconnected servers and also on interconnected edgepoints. Moreover, Abrams' and Lanzillo's inventions do not teach a method for the replication of the resource on a requesting host (target client of Abrams).

Further, according to the present invention, all the replicates of a particular resource are connected together. The connection between the symbionts encapsulating the resource in the network provides load information about neighboring symbionts

(hereinafter referred to as connected replicates). This reduces the time taken by a symbiont to make a decision about handling a request, for example, if the symbiont retains the information about the connected replicates, and redirects the request to a connected replicate with the least load. The support for these recitations is found on page 11, lines 9-24. However, Abrams does not teach any connection between the object replicas, it teaches that thousands of servers in hundreds of edgepoints are globally deployed and linked to create a virtual single server view throughout the world. Further, Abrams teaches about the intelligence of the GD, to route the requests to an optimal edgepoint. Lanzillo teaches that a plurality of MCs are connected to create a LAN, wherein only one PMC is the decision-making component routing user requests to other MCs that are interconnected in the LAN. Whereas, in the present invention, each symbiont is an independent decision-making component that redirects, replicates and exits from the multiply connected ring and can run in any type of network.

Independent claims 20 and 36

As to claims 20 and 36, Abrams teaches a method and a system for regulating resources in a client-server architecture. However, the present invention defines hosts, symbionts and resources that are not limited to the client-server architecture. For example, the present invention can be practiced on a peer-to-peer architecture in a LAN environment. Further, a WAN or a MAN environment may also be used to enable the invention. The only requirements for the enablement of the invention are software programs such as the symbionts and the hosts.

The fact that the resources are connected in a multiply connected ring makes the network of resources more robust. The arrangement also facilitates better sharing of information among connected replicates. All the symbionts present in the network have information pertaining to all the available resources. Further, the multiply connected ring makes the symbiont architecture scalable. Thus, if the load on a symbiont increases above a threshold value (l_{max}), the symbiont replicates. Similarly, the symbiont ceases to exist if the load on it falls below a minimum threshold value (l_{min}), thereby keeping the number of the symbionts at an optimum level. The support for these recitations is found on page 10, lines 12-29.

Further, the present invention teaches the use of two different thresholds. However, Abrams and Lanzillo do not teach the use of two thresholds, l_{max} and l_{min}, or any relation between them. Moreover, Lanzillo teaches that when a designated PMC process is stopped for any reason, one of the other MCs automatically takes over the PMC's role. However, in the present invention, every symbiont is an independent PMC.

Dependent claim 27 has been canceled without prejudice.

Dependent claims 31 and 44

As to claims 31 and 44, the present invention elaborates further on the process of keeping at least one instance of a resource in a network of computers. One of the symbionts is marked immortal so that the intelligence required for handling a request for a resource is never deleted. Therefore, instead of maintaining an entire hardware (the appshot in the servers of Abrams), the present invention teaches the use of keeping only

a single symbiont alive as a means to carry forward the intelligence required to handle a request. The support for this recitation is found on page 13, lines 21-24. Further, Lanzillo's invention does not teach about the immortal resource.

Dependent claims 32 and 45

In claims 32 and 45, it is recited that a symbiont may cease to exist if the load on the symbiont falls below a predetermined threshold l_{min}. The load of the symbiont is checked at regular time intervals. Further, it is ensured that these intervals are not kept too short, to avoid churning, i.e., the symbionts do not keep dying and taking birth at a high frequency. This ensures that the symbiont population is kept at an optimum level. The support for these recitations is found on page 13, lines 16-19. However, Abrams teaches the use of an appshot to maintain the ideal number of replicas. Further, Abrams and Lanzillo neither teaches the use of the predetermined threshold l_{min} nor checks the load of the connected replicates (Abrams' edge serves and Lanzillo's MCs).

Dependent claims 33–34 and 46-47

In claims 33 and 34, it is recited that that the time interval for a symbiont to check its own load depends on the load on it. Therefore, the time intervals depend on the actual load variations on the symbiont, measured in real time. The predetermined threshold l_{min} is directly dependent on the number of symbionts in the multiply connected ring. Further, Abrams and Lanzillo do not teach the use of the predetermined threshold l_{min}. According to Abrams, the total amount of processing capacity required is equal to the maximum aggregated demand in any time period, which makes the system time-dependent.

However, the present invention is neither time-dependent nor is there any limit on the number of symbionts.

Dependent claims 35 and 48 have been canceled without prejudice.

The present claims have been amended to highlight the distinctions of the present invention over the prior art, and it is respectfully submitted that the claims are now clearly patentable over the art of record, and notice to that effect is earnestly solicited. If the Examiner has any questions regarding this matter, the Examiner is requested to telephone the applicants' attorney at the numbers listed below, prior to issuing a further Office Action.

Respectfully Submitted,

By

William L. Botjer
Reg. No. 27,990
PO Box 478
Center Moriches, NY 11934
(212) 737-5728 (Tue-Thurs)
(631) 874-4826 (Mon & Fri)
(631) 834-0611 (cell if others busy)

Dated:September 29, 2006